Route map to Programmable Components

INSPIRATIONAL CURRICULUM SUPPORT, ADVICE & TRAINING
www.icsat.co.uk

Waypoints				
Electrical/Electronic circuits (Passive)	**Project ideas**	**Science**	**Computing**	**KS**
• Lights/LEDs (Std & flashing) • Switches • Bells/Buzzers • Motors • Resistors • Simple circuit theory (Calcs at KS3) • PCBs (Pre made only) • Copper tape	• Dixons - Night light • ICSAT - Animated signage • ICSAT Torch • ICSAT Roundabout • ICSAT Automata • ICSAT Dusk Lights on!	Construct a simple series electrical circuit, identifying and naming its basic parts, including cells, wires, bulbs, switches, buzzers and motors	Use of simple coding for control of product, flowchart based eg. Flowol, Go PC, Scratch Links between D&T and Computing	**Lower 2**
Electronic circuits (Active)	**Project ideas**	**Science**	**Computing**	**KS**
All the above plus: • Transistor (Switching only) • Logic gates (NOT, AND, OR) • Sensors 1. LDR 2. Thermistor 3. Pressure 4. Position 5. Magnetic sensor • Circuit simulation • Circuit design (CADCAM) • PCBs (Design & make) • Breadboard (Prototyping) • Veroboard (Prototyping)	• GHS - Torch (Thyristor) • GHS - Night light (auto lighting) • ICSAT - Personal alarm • Dixons - Night light+ • ICSAT - Touch Torch • ICSAT - Touch Light/Lamp • ICSAT Dusk Lights on! • ICSAT Joule Thief Lighting	Compare and give reasons for variations in how components function, including the brightness of bulbs, the loudness of buzzers and the on/off position of switches. Use recognised symbols when representing a simple circuit in a diagram. Designing and making a set of traffic lights, a burglar alarm or some other useful circuit.	Use of simple coding for control of product, flowchart based eg. Flowol, Go PC, Scratch Top end 2 and into 3, use of Python (Textual) Use of Arduino, RPI for coding and control & monitoring Links between D&T and Computing to develop (KS3)	**Upper 2 & Lower 3**
Configurable electronics	**Project ideas**	**Science**		**KS**
All the above plus: • Timers (Astable / Monostable) • Capacitor & Resistor (for timing) - mathematical modelling • Modules/Pre-programmed IC's (with a number of functions) • LEDs (RGB, lighting strips, 7 segment) - resistor calcs/maths modelling • Loudspeakers / Piezo transducers	• GHS - Night light (Timed) • ICSAT - Game in a box (PIC dice) • ICSAT - 'Help me' (POV) • Dixons - Monsters (eTextiles) • Dixons USB lamp • ICSAT - Music box (Simple) • ICSAT - eCard (Greetings) • ICSAT – Spin-It (8 games) • ICSAT – Money Box	Electric current, measured in amperes, in circuits, series and parallel circuits, currents add where branches meet and current as flow of charge Potential difference, measured in volts, battery and bulb ratings; resistance, measured in ohms, as the ratio of potential difference (p.d.) to current. Differences in resistance between conducting and insulating components (quantitative). The magnetic effect of a current, electromagnets, D.C. motors (principles only).		**Mid 3**
Programmable electronics	**Project ideas**		**Computing**	**KS**
All of the above plus • PICs - Starter (Can be limited) ○ PICAXE / Pixie ○ GENIE / Pixie ○ Crumble • SBC's - advanced (Easier & flexible) ○ Arduino ○ Parallax FLIP ○ BBC micro:bit ○ RPI • LEDs (RGB, Neopixels) • LCDs (Displays) • Motors (Servo, stepper) • Modules (such as): ○ Accelerometers / GPS ○ WiFI / Bluetooth ○ ADCs, DACs & audio ○ SD card readers ○ IR sensors ○ RFID sensors & tags ○ Ultra sonic distance sensor ○ Proximity sensor (PIR)	• ICSAT - Board game (Active board) • ICSAT - Animated signage • ICSAT - Active Toy • ICSAT - Play mat (eTextiles) • ICSAT - Music box (Adv functions) • ICSAT - Rescue Me (POV) • ICSAT – Rescue Me (eTextiles) • ICSAT - Moody Lights • ICSAT - Musical Toy • ICSAT - Activity Toy • ICSAT - Animatronic Creature • ICSAT – Safety clothing (eTextiles) • ICSAT - Smart Clothing (eTextiles) • ICSAT – GoPro Panoramic drive • ICSAT - GoPro Star drive		Use of advanced coding for control and embedding intelligence into a product, textual based eg Basic, Python, C Use of PIC's, Arduino, FLIP, RPI etc. Links between D&T and Computing to develop	**Upper 3 & Lower 4**

Notes

There is no requirement at KS1 for Electronics/Programmable Components, but some use only via the Computing Route for simple control.
Lower KS2 = Y2 & 3. Upper KS2 = Y5 & 6, Lower KS3 = 7, Mid KS3 = 8, Upper KS3 = 9, Lower 4 = KS4 (GCSE Y10)

Programming map for Programmable Components

Programming Waypoints for D&T			
Algorithms	**KS2**	**KS3**	**GCSE**
Use a systematic approach to problem solving and algorithm creation representing those algorithms using pseudo-code and flowcharts	√	√	√
Programming			
Be able to write programs in a high-level programming language:			
• Scratch / Blockly	√	√	√
• Python, BASIC	X	√	√
• C++,Java	X	X	√
Understand and use the following appropriately:			
• integer	√	√	√
• Boolean	X	X	√
• Character, string	X	√	√
Understand and use the following statement types can be combined in programs:			
• variable & constant declaration	X	√	√
• assignment [x=10...]	√	√	√
• iteration [for..next, Do..loop, while....loop]	√	√	√
• selection [if..then, select..case]	√	√	√
• subroutine (procedure/function) [gosub, call...]	X	√	√
Use meaningful identifier names and know why it is important to use them	√	√	√
Arithmetic operations			
Understand and use, arithmetic operators:			
• add [+], subtract [-], divide [/], multiply [*]	√	√	√
• Shifts (left & right)	X	√	√
Be able to use random number generation	√	√	√
Relational operations			
Understand and use, relational operators:			
• equal to [=], less than [<], greater than [>], not equal to [<>]	√	√	√
• less than or equal to [<=], greater than or equal to [>=]	X	√	√
Logic operators			
Understand and use, logic operators:			
• AND, OR, NOT	X	√	√
Understand and use, bitwise logic operators: AND, OR [Masking& merging]	X	X	√
Data structures			
Understand and use arrays or Data tables, using one or two-dimensions as needed	X	√	√
Input/Output			
Write code that accepts and responds appropriately to inputs from input devices & sensors	√	√	√
Write code to scale analogue values into useful ranges	X	√	√
Write code that sends data to output devices	√	√	√
String operations			
Write code that uses strings for displays devices [LCD displays]	X	√	√
Subroutines (procedures / functions)			
Understand and use subprograms and be able to write code that uses user-written and pre-existing (built-in, libraries) subprograms	X	√	√
Be able to create subprograms that use in & out parameters	X	X	√
Be able to use local variables	X	X	√
Structured programming			
Be able to use a structured approach to Programming, including:			
• modularised programming	X	√	√
• (local variables, parameters) and return values	X	X	√
Number bases			
Understand that computers use binary patterns to represent different types of data including text, image, sound and integer and instructions	√	√	√
Understand and use binary to represent decimal values between 0 and 255 in binary.	X	√	√
Be able to convert in both directions between binary and decimal ***8 bits only***	X	√	√
Computer Systems			
Understand the terms hardware and software and the relationship between them	√	√	√
Understand the term 'embedded system' and explain how an embedded system differs from a non-embedded system, using examples they are familiar with	X	√	√

Notes

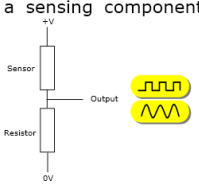
Items highlighted are D&T specific for advanced programming

Programmable Components overview for KS3 and GCSE Design and Technology

INPUT

Most electronic systems use one or more inputs, so the electronics can respond to user inputs, environmental conditions, mechanical and/or electronic events.

Virtually all input devices are a sensor of some type, basic sensors all have the same arrangement of a sensing component and a resistor:



Some sensor produce a **Digital (on/off)** signal, other produce an **Analogue (variable)** signal.

Digital sensors
- PTM switch
- PTB switch
- Tilt switch
- Moisture probe
- Magnetic switch
- Photodiode

Off = 0V, On = +V

Analogue sensors
- LDR
- Thermistor
- Piezo transducer
- Potentiometer

The signal can vary between 0V and +V

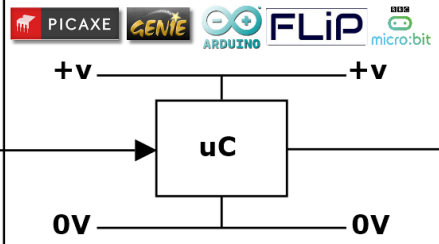
Some sensors are chip based and need additional electronics and/or coding to make them function, such as temperature, humidity, PIR, rotary position, GPS, accelerometers etc.

PROCESS

The key process block for all modern electronics is the **Microcontroller (uC)**.

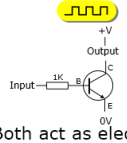
This electronics is known as **Embedded Electronics,** since the microcontroller is 'embedded' into a product.

In school's the most common microcontroller systems are **PICAXE, Genie, Micro:bit, Crumble, and Arduino**.



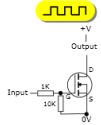
The other useful component is the **transistor**, 2 types are useful for us:

NPN



Transistor:
Off < 0.7V
On >0.7V

MOSFET



Transistor:
Off < 3.0V
On >3.0V

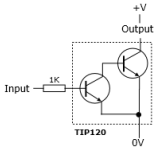
Both act as electronic switches

INTERFACE

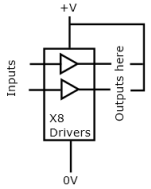
The output from a microcontroller can only supply about **10mA at maximum**, if more current is needed an **Interface driver** will be needed.

Single transistor drivers using either an **NPN** or **MOSFET** version, usually for a **current <250mA.**

For **currents >250mA** a Darlington Driver is required.



Sometimes using a microcontroller more drivers might be needed, the best solution here is to use a driver chip that contains 7 or 8 drivers.



OUTPUT

Output devices all fall into one of the following groups:

Light:
- Single LED
- Bi coloured LED
- Tri coloured LED
- LED Bars
- 7 segment LED
- RGB LED
- NeoPixels - needs coding to operate

Audio:
- Piezo Transducer
- Buzzer
- Speaker - needs a transistor driver

Motors:
- DC motor - needs a transistor driver (on/off)
- DC motor - motor driver (CW, CCW, off)
- Servo motor - 180° - needs coding but no driver
- Servo motor - 360° - needs coding but no driver
- Stepper motor - 360° - needs coding & driver chip
- Solenoid - linear movement - needs a transistor driver

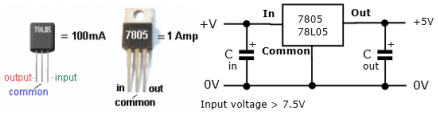
For further information see the full set of individual eCards covering Inputs, Processes, Interfaces, Outputs and Power

POWER

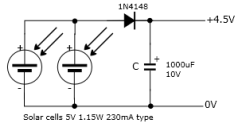
All electronic systems require a power supply, the most common solutions are:
- Batteries - check voltage, capacity & size
- Solar Cells - check voltage & current
- Super capacitors - check voltage & size
- USB power - +5V

You will need to select the most appropriate for you solution taking into account it's **use and it's voltage & current requirements**.

For microcontrollers you may need to use a **voltage regulator** to ensure the correct voltage is used, normally +5V, if you are using a 9V battery for example



Solar cells can provide a reasonable supply, but you often have to use more than 1 cell to get a higher enough voltage & current.



Super Capacitors can charged up via another battery, solar cells or a USB connection. They can power a low current circuit for up to 15 mins. They make good power sources for portable lighting solutions.